Claims

What is claimed is :

1. a method for executing structured symbolic machine code on a microprocessor,
where said microprocessor is part of a data processing system containing a memory system,
where said memory system is defined to have a memory hierarchy containing :
    - one or more register files of said microprocessor
    - one or more data caches at different memory hierarchy levels
    - a main memory

where said microprocessor is able to perform speculative branch prediction,
where said speculative branch prediction is based on a branch history which may store outcomes of branches which are not yet resolved at the point in time when a branch prediction is being made,
where unresolved branch outcomes may update counter states within the branch history,
where said counter states may concern counter states stored in a pattern history table,
where said microprocessor has an instruction set containing one or more instructions of which one or more operands and/or results may specify one or more symbolic variables,
where said structured symbolic machine code contains one or more regions, where at least one (R1) of said regions contains symbolic machine code containing one of or both of the following information :
    - information (IF1)
    - information (IF2)

where said region (R1) may further contain one or more of the following information :
    - information (IF3)
    - information (IF4)
    - information (IF5)
    - information (IF6)

where said information (IF1) to (IF6) may be stored into one or more dedicated memories within the microprocessor during the execution of said structured symbolic machine code,

where the symbolic machine contained in each of said regions contains one or more instructions of which one or more operands and/or results specify one or more symbolic variables,

where each of said symbolic variables specifies one or more entries (Es) of a memory other than a register file of said microprocessor, where said entries (Es) are used by the microprocessor in order to determine the addresses within the memory hierarchy where the values of said symbolic variables may or may have to be stored to and/or loaded from during execution of said structured symbolic machine code,

where said information (IF1) contains :

- a symbolic variable (S1) of said region (R1)
- a definition address of said symbolic variable (S1)

where said information (IF1) may further contain one or more of the following information :

- the type of said symbolic variable (S1)
- one or more labels specifying each a dependence group which said symbolic variable (S1) pertains to
- the lexicographical order of said symbolic variable (S1)
- an address variable associated with said symbolic variable (S1)
- a flag which indicates whether said definition address is fixed during machine code execution or not
- the WAR-lexicographical order of said symbolic variable (S1)
- the RAW-lexicographical order of said symbolic variable (S1)
- the WAW-lexicographical order of said symbolic variable (S1)

where said information (IF2) contains a symbolic constant (C1) of said region (R1) and one or both of the following information :

- the type of said symbolic constant (C1)
- the value of said symbolic constant (C1)

where said information (IF3) contains :

- a node (N1) of a branch tree associated with said region (R1)
- one or more successor nodes of said node (N1)

where said information (IF3) may further contain one or more of the following information :

- a symbolic variable associated with node (N1) and which is used as a predication variable of a predicated instruction of said region (R1)
- a branch address associated with said node (N1)
- a branch address associated with a successor node of said node (N1)
- the number of instructions associated with said node (N1)
- none, one or more branch counts associated with said node (N1), where said branch counts may be used to do a correct updating of counter states of said pattern history table after branch miss-predictions have occurred, and where each of said branch counts may correspond to the counter state of a counter within the pattern history table at the point in time when or before the prediction of the value (V1) of a predication

variable associated with said node (N1) shall start or shall be done and where said counter state may be used in the prediction of said value (V1)

- a branch flag associated with said node (N1), where said branch flag indicates whether there is just one successor node of said node (N1) or not
- a branch-back-flag associated to a successor node of said node (N1), where said branch-back-flag indicates whether the following two conditions are satisfied or not :
    1. said successor node belongs to the same region as said node (N1)
    2. there exists a branch path through the region from said successor node to said node (N1)
- a region-flag associated to a successor node of said node (N1), where said region-flag indicates whether said successor node belongs to the same region than said node (N1) or not
- a function-flag associated to a successor node of said node (N1), where said function-flag indicates whether the following two conditions are satisfied or not :
    1. said successor node belongs to another region (rx) than said node (N1)
    2. said region (rx) contains machine code which is associated to the source code of a function declaration or definition
- an exit-flag associated to a successor node of said node (N1), where said exit-flag indicates whether the following two conditions are satisfied or not :
    1. said successor node belongs to another region than said node (N1)
    2. the branch address associated to said successor node is not known statically before runtime and may be determined dynamically by the microprocessor during machine code execution

where said microprocessor may dynamically generate, during machine code execution, one or more branch counts associated with said node (N1), by setting each of said branch counts equal to the counter state of a counter within the pattern history table at the point in time when or before the prediction of the value (V3) of a predication variable associated with said node (N1) shall start or shall be done, where said branch counts may be used to do a correct updating of counter states of said pattern history table after branch miss-predictions have occurred, and where said counter states may be used in the prediction of said value (V3),

where said information (IF4) contains :

- a value (V2) obtained by concatenating the values and/or the labels of one or more predication variables stored in said branch history table at some point in time during machine code execution
- an address (A1) associated with said value (V2), where said address (A1) is the address of a specific instruction pertaining to said region (R1) and where said address (A1) may correspond to the start address of a trace within said region (R1)

where said information (IF5) contains :

- a symbolic variable (S2) of said region (R1)
- one or more labels specifying each a dependence group which the symbolic variable (S2) pertains to

where said information (IF6) is associated with said region (R1) and allows the microprocessor to :

- identify said region (R1) among several other regions
- determine the location of the before mentioned information (IF1) to (IF5) within the address space of the microprocessor
- determine a program counter value which is associated with said region (R1)
- determine which part of the information (IF1) to (IF5) is common both to said region (R1) and to a region other than region (R1)

2. a microprocessor for executing structured symbolic machine code,
where said microprocessor is part of a data processing system containing a memory system,
where said memory system is defined to have a memory hierarchy containing :
   - one or more register files of said microprocessor
   - one or more data caches at different memory hierarchy levels
   - a main memory

   where said microprocessor is able to perform speculative branch prediction,
where said speculative branch prediction is based on a branch history which may store outcomes of branches which are not yet resolved at the point in time when a branch prediction is being made,
where unresolved branch outcomes may update counter states within the branch history,
where said counter states may concern counter states stored in a pattern history table,
where said microprocessor has an instruction set containing one or more instructions of which one or more operands and/or results may specify one or more symbolic variables,

   where said structured symbolic machine code contains one or more regions, where at least one (R2) of said regions contains symbolic machine code containing one or both of the following information :
   - information (IF7)
   - information (IF8)

   where said region (R2) may further contain one or more of the following information :
   - information (IF9)
   - information (IF10)
   - information (IF11)
   - information (IF12)

where said information (IF7) to (IF12) may be stored into one or more dedicated memories within the microprocessor during the execution of said structured symbolic machine code,

where the symbolic machine contained in each of said regions contains one or more instructions of which one or more operands and/or results specify one or more symbolic variables,

where each of said symbolic variables specifies one or more entries (Et) of a memory other than a register file of said microprocessor,

where said entries (Et) are used by the microprocessor in order to determine the addresses within the memory hierarchy where the values of said symbolic variables may or may have to be stored to and/or loaded from during execution of said structured symbolic machine code,

where said information (IF7) contains a symbolic variable (S5) of said region (R2), said information (IF7) being used by said microprocessor to determine a definition address of said symbolic variable (S5), where said information (IF7) may further be used by said microprocessor to determine one or more of the following :

- the type of said symbolic variable (S5)
- one or more labels specifying each a dependence group which said symbolic variable (S5) pertains to
- the lexicographical order of said symbolic variable (S5)
- an address variable associated with said symbolic variable (S5)
- a flag which indicates whether said definition address is fixed during machine code execution or not
- the WAR-lexicographical order of said symbolic variable (S5)
- the RAW-lexicographical order of said symbolic variable (S5)
- the WAW-lexicographical order of said symbolic variable (S5)

where said information (IF8) contains a symbolic constant (C2) of said region (R2), said information (IF8) being used by said microprocessor to determine the type and/or the value of said symbolic constant (C2),

where information (IF9) contains :

- a node (N2) of a branch tree associated with said region (R2)
- one or more successor nodes of said node (N2)

where said information (IF9) may further be used by said microprocessor to determine one or more of the following :

- a symbolic variable associated with said node (N2) and which is used as a predication variable of a predicated instruction of said region (R2)

- a branch address associated with said node (N2)
- a branch address associated with a successor node of said node (N2)
- the number of instructions associated with said node (N2)
- one or more branch counts associated with said node (N2), where said branch counts may be used to do a correct updating of counter states of said pattern history table after branch miss-predictions have occurred, and where each of said branch counts may correspond to the counter state of a counter within the pattern history table at the point in time when or before the prediction of the value (V4) of a predication variable associated with said node (N2) shall start or shall be done and where said counter state may be used in the prediction of said value (V4)
- a branch flag associated with said node (N2), where said branch flag indicates whether there is just one successor node of said node (N2) or not
- a branch-back-flag associated to a successor node of said node (N2), where said branch-back-flag indicates whether the following two conditions are satisfied or not :
  1. said successor node belongs to the same region as said node (N2)
  2. there exists a branch path through the region from said successor node to said node (N2)
- a region-flag associated to a successor node of said node (N2), where said region-flag indicates whether said successor node belongs to the same region than said node (N2) or not
- a function-flag associated to a successor node of said node (N2), where said function-flag indicates whether the following two conditions are satisfied or not :
  1. said successor node belongs to another region (rx) than said node (N2)
  2. said region (rx) contains machine code which is associated to the source code of a function declaration or definition
- an exit-flag associated to a successor node of said node (N2), where said exit-flag indicates whether the following two conditions are satisfied or not :
  1. said successor node belongs to another region than said node (N2)
  2. the branch address associated to said successor node is not known statically before runtime and may be determined dynamically by the microprocessor during machine code execution

where said microprocessor may dynamically generate, during machine code execution, one or more branch counts associated with said node (N2) by setting each of said branch counts equal to the counter state of a counter within the pattern history table at the point in time when or before the prediction of the value (V6) of a predication variable associated with said node (N2) shall start or shall be done and by storing said branch counts in some dedicated memory, where said branch counts may be used to do a correct updating of counter states of said pattern history table after branch miss-predictions have occurred, and where said counter states may be used in the prediction of said value (V6),

where said information (IF10) contains a value (V5) obtained by concatenating the values and/or the labels of one or more predication variables stored in said branch history table at some point in time during machine code execution, said information (IF10) being used by said microprocessor to determine an address (A2) associated with said value (V5), where said address (A2) is the address of a specific instruction pertaining to said region (R2) and where said address (A2) may correspond to the start address of a trace within said region (R2),

where said information (IF11) contains a symbolic variable (S6) of said region (R2) and where said symbolic variable (S6) may correspond to a pointer variable declared in a source code program associated with said region (R2), said information (IF11) being used by said microprocessor to determine one or more labels specifying each a dependence group which the symbolic variable (S6) pertains to

where said information (IF12) is associated with said region (R2) and allows the microprocessor to :

- identify said region (R2) among several other regions
- determine the location of the before mentioned information (IF7) to (IF11) within the address space of the microprocessor
- determine a program counter value which is associated with said region (R2)

determine which part of the information (IF7) to (IF11) is common both to said region (R2) and to a region other than region (R2)

3. A method as claimed in claim 1.,

where said region (R1) contains symbolic machine code containing one of or both of the following information :

- information (IF1)
- information (IF2)

where said region (R1) further contains one or more of the following information :

- information (IF3)
- information (IF4)
- information (IF5)
- information (IF6)

4. A microprocessor as claimed in claim 2.,

where said region (R2) contains symbolic machine code containing one of or both of the following information :

- information (IF7)
- information (IF8)

where said region (R1) further contains one or more of the following information :

- information (IF9)
- information (IF10)
- information (IF11)
- information (IF12)

5. A method as claimed in claim 3.,

where said region (R1) contains symbolic machine code containing said information (IF1), (IF2), (IF3), (IF4), (IF5) and (IF6),

where information (IF1) is a table (T1) which is indexed by the labels of one or more symbolic variables of said region (R1), and where each entry (E1) of said table (T1) contains the following information :

- the type of the symbolic variable specified by said entry (E1)
- a definition address of the symbolic variable specified by said entry (E1)
- one or more labels specifying each a dependence group which the symbolic variable specified by said entry (E1) pertains to
- an address variable associated with the symbolic variable specified by said entry (E1)
- a flag which indicates whether said definition address of the symbolic variable specified by said entry (E1) is fixed during machine code execution or not

where information (IF2) is a table (T2) indexed by the labels of one or more symbolic constants of said region (R1) and where each entry (E2) of said table (T2) contains the type and/or the value of a symbolic constant specified by said entry (E2),

where information (IF3) is a table (T3) indexed by labels of one or more nodes of a branch tree associated with said region (R1) and where each entry (E3) of table (T3) contains the following information :

- one or more successor nodes of the node specified by said entry (E3)
- a symbolic variable (S9) associated with the node specified by said entry (E3), where said symbolic variable (S9) is a predication variable of a predicated instruction of said region (R1)
- a branch address associated with the node specified by said entry (E3)

- a branch address associated with a successor node of the node specified by said entry (E3)
- the number of instructions associated with the node specified by said entry (E3)
- none, one or more branch counts associated with the node specified by said entry (E3), where said branch counts may be used to do a correct updating of counter states of said pattern history table after branch miss-predictions have occurred, and where each of said branch counts may correspond to the counter state of a counter within the pattern history table at the point in time when or before the prediction of the value (V9) of said predication variable (S9) shall start or shall be done and where said counter state may be used in the prediction of said value (V9)
- a branch flag associated with said entry (E3), where said branch flag indicates whether there is just one successor node of the node specified by said entry (E3) or not
- a branch-back-flag associated to a successor node of the node specified by said entry (E3), where said branch-back-flag indicates whether the following two conditions are satisfied or not :
  1. said successor node belongs to the same region as the node specified by said entry (E3)
  2. there exists a branch path through the region from said successor node to the node specified by said entry (E3)
- a region-flag associated to a successor node of the node specified by said entry (E3), where said region-flag indicates whether said successor node belongs to the same region than the node specified by said entry (E3) or not
- a function-flag associated to a successor node of the node specified by said entry (E3), where said function-flag indicates whether the following two conditions are satisfied or not :
  1. said successor node belongs to another region (rx) than the node specified by said entry (E3)
  2. said region (rx) contains machine code which is associated to the source code of a function declaration or definition
- an exit-flag associated to a successor node of the node specified by said entry (E3), where said exit-flag indicates whether the following two conditions are satisfied or not :
  1. said successor node belongs to another region than the node specified by said entry (E3)
  2. the branch address associated to said successor node is not known statically before runtime and may be determined dynamically by the microprocessor during machine code execution

where said microprocessor may dynamically generate, during machine code execution, one or more branch counts associated with said node specified by said entry (E3) by setting each of said branch counts equal to the counter state of a counter within the pattern history table at the point in time when or before the prediction of the value (Vp) of a predication variable associated with said entry (E3) shall start or shall be done and by storing said branch counts in some dedicated memory, where said branch counts may be used to do a correct updating of counter states of said pattern history table after branch miss-predictions have occurred, and where said counter states may be used in the prediction of said value (Vp),

where information (IF4) is a table (T4) indexed by a value (V10) obtained by concatenating the values and/or the labels of one or more predication variables stored in said branch history table at some point in time during machine code execution, and where each entry of said table (T4) stores an address (A3) associated with said value (V10), where said address (A3) is the address of a specific instruction pertaining to said region (R1) and where said address (A3) may correspond to the start address of a trace within said region (R1),

where information (IF5) is a table (T5) indexed by the labels of one or more symbolic variables of said region (R1), where said symbolic variables may refer to pointer variables declared in a source code program associated with said region (R1) and where each entry (E4) of said table (T5) contains one or more labels specifying each a dependence group which the symbolic variable specified by said entry (E4) pertains to

6. A microprocessor as claimed in claim 4.,
where said region (R2) contains symbolic machine code containing said information (IF7), (IF8), (IF9), (IF10) , (IF12) and (IF12),
where information (IF7) is a table (T7) which is indexed by the labels of one or more symbolic variables of said region (R2), and where each entry (E5) of said table (T7) contains the following information :
   - the type of the symbolic variable specified by said entry (E5)
   - a definition address of the symbolic variable specified by said entry (E1)
   - one or more labels specifying each a dependence group which the symbolic variable specified by said entry (E5) pertains to
   - an address variable associated with the symbolic variable specified by said entry (E5)
   - a flag which indicates whether said definition address of the symbolic variable specified by said entry (E5) is fixed during machine code execution or not

where information (IF8) is a table (T8) indexed by the labels of one or more symbolic constants of said region (R2) and where each entry (E6) of said table (T8) contains the type and/or the value of a symbolic constant specified by said entry (E6),

where information (IF9) is a table (T9) indexed by labels of one or more nodes of a branch tree associated with said region (R2) and where each entry (E7) of table (T9) contains the following information :

- one or more successor nodes of the node specified by said entry (E7)
- a symbolic variable (S10) associated with the node specified by said entry (E7), where said symbolic variable (S10) is a predication variable of a predicated instruction of said region (R2)
- a branch address associated with the node specified by said entry (E7)
- a branch address associated with a successor node of the node specified by said entry (E7)
- the number of instructions associated to the node specified by said entry (E7)
- none, one or more branch counts associated with the node specified by said entry (E7), where said branch counts may be used to do a correct updating of counter states of said pattern history table after branch miss-predictions have occurred, and where each of said branch counts may correspond to the counter state of a counter within the pattern history table at the point in time when or before the prediction of the value (V11) of said predication variable (S10) shall start or shall be done and where said counter state may be used in the prediction of said value (V11)
- a branch flag associated with said entry (E7), where said branch flag indicates whether there is just one successor node of the node specified by said entry (E7) or not
- a branch-back-flag associated to a successor node of the node specified by said entry (E7), where said branch-back-flag indicates whether the following two conditions are satisfied or not :
    1. said successor node belongs to the same region as the node specified by said entry (E7)
    2. there exists a branch path through the region from said successor node to the node specified by said entry (E7)
- a region-flag associated to a successor node of the node specified by said entry (E7), where said region-flag indicates whether said successor node belongs to the same region than the node specified by said entry (E7) or not
- a function-flag associated to a successor node of the node specified by said entry (E7), where said function-flag indicates whether the following two conditions are satisfied or not :
    1. said successor node belongs to another region (rx) than the node specified by said entry (E7)

2. said region (rx) contains machine code which is associated to the source code of a function declaration or definition

- an exit-flag associated to a successor node of the node specified by said entry (E7), where said exit-flag indicates whether the following two conditions are satisfied or not :
  1. said successor node belongs to another region than the node specified by said entry (E7)
  2. the branch address associated to said successor node is not known statically before runtime and may be determined dynamically by the microprocessor during machine code execution

where said microprocessor may dynamically generate, during machine code execution, one or more branch counts associated with said node specified by said entry (E7) by setting each of said branch counts equal to the counter state of a counter within the pattern history table at the point in time when or before the prediction of the value (Vq) of a predication variable associated with said entry (E7) shall start or shall be done and by storing said branch counts in some dedicated memory, where said branch counts may be used to do a correct updating of counter states of said pattern history table after branch miss-predictions have occurred, and where said counter states may be used in the prediction of said value (Vq),

where information (IF10) is a table (T10) indexed by a value (V12) obtained by concatenating the values and/or the labels of one or more predication variables stored in said branch history table at some point in time during machine code execution, and where each entry of said table (T10) stores an address (A4) associated with said value (V12), where said address (A4) is the address of a specific instruction pertaining to said region (R2) and where said address (A4) may correspond to the start address of a trace within said region (R2),

where information (IF11) is a table (T11) indexed by the labels of one or more symbolic variables of said region (R2), where said symbolic variables may refer to pointer variables declared in a source code program associated with said region (R2) and where each entry (E8) of said table (T11) contains one or more labels specifying each a dependence group which the symbolic variable specified by said entry (E8) pertains to

7. A method as claimed in claim 1.,
   where one or more instructions of the instruction set of said microprocessor may have an instruction format containing a bit-field (B10) which indicates :
   - whether the value of an instruction operand is stored in a register file or not
   - or whether the value of an instruction operand has to be written into a register file or not
   - or whether the value of an instruction result has to be written into a register file or not

and where said instruction operand and said instruction result specify a symbolic variable, and where the meaning of the bit-field (B10) may depend on the meaning of other bit-fields of the instruction format of any instruction of said instruction set,

where one or more instructions of the instruction set of said microprocessor may have an instruction format containing a bit-field (B11) which indicates whether a symbolic variable specified by an operand and/or result of an instruction (I7) is specified by an operand and/or result of another instruction (I8) or not, and where instruction (I8) may enter or may have to enter a certain pipeline stage either before or after instruction (I7) enters or is allowed to enter a certain pipeline stage, where the meaning of the bit-field (B11) may depend on the meaning of other bit-fields of the instruction format of any instruction of said instruction set,

where one or more instructions of the instruction set of said microprocessor may have an instruction format containing a bit-field (B13) which indicates whether a symbolic variable specified by an instruction operand and/or an instruction result is data dependent or not,
where the meaning of the bit-field (B13) may depend on the meaning of other bit-fields of the instruction format of any instruction of said instruction set,

where at least one instruction (I9) of the instruction set of said microprocessor may have an instruction format containing a bit-field (B12) which specifies an instruction (I10), and where instruction (I10) may have to enter a certain pipeline stage either before or after instruction (I9) enters or is allowed to enter a certain pipeline stage,
where the meaning of the bit-field (B12) may depend on the meaning of other bit-fields of the instruction format of any instruction of said instruction set,

where said microprocessor may have an instruction which, when it enters a certain pipeline stage, indicates that certain symbolic variables (Sk) are no longer used and that the values of symbolic variables (Sk) may be overwritten

where one or more instructions of the instruction set of said microprocessor may have an instruction format containing a bit-field (B14) which specifies whether or not a symbolic variable specified as instruction result may be allocated to the same register as the one to which one of the instruction operands is allocated

where one or more instructions of the instruction set of said microprocessor may have an instruction format containing a bit-field (B15) which specifies whether or not the value of a symbolic variable (sv4) specified as instruction result is the value of the definition address of said symbolic variable (sv4) where one or more instructions of the instruction set of said microprocessor may have an instruction format containing a bit-field (B16) which specifies whether or not the value of a symbolic variable specified as instruction result is the value of a predication variable

where one or more instructions of the instruction set of said microprocessor may have an instruction format containing a bit-field (B17) which specifies whether or not the value of a symbolic variable specified as instruction result corresponds to a branch address

8. A microprocessor as claimed in claim 2.,

where one or more instructions of the instruction set of said microprocessor may have an instruction format containing a bit-field (B10) which indicates :

- whether the value of an instruction operand is stored in a register file or not
- or whether the value of an instruction operand has to be written into a register file or not
- or whether the value of an instruction result has to be written into a register file or not

and where said instruction operand and said instruction result specify a symbolic variable, where the meaning of the bit-field (B10) may depend on the meaning of other bit-fields of the instruction format of any instruction of said instruction set,

where one or more instructions of the instruction set of said microprocessor may have an instruction format containing a bit-field (B11) which indicates whether a symbolic variable specified by an operand and/or result of an instruction (I7) is specified by an operand and/or result of another instruction (I8) or not, and where instruction (I8) may enter or may have to enter a certain pipeline stage either before or after instruction (I7) enters or is allowed to enter a certain pipeline stage, where the meaning of the bit-field (B11) may depend on the meaning of other bit-fields of the instruction format of any instruction of said instruction set,

where one or more instructions of the instruction set of said microprocessor may have an instruction format containing a bit-field (B13) which indicates whether a symbolic variable specified by an instruction operand and/or an instruction result is data dependent or not,
where the meaning of the bit-field (B13) may depend on the meaning of other bit-fields of the instruction format of any instruction of said instruction set,

where at least one instruction (I9) of the instruction set of said microprocessor may have an instruction format containing a bit-field (B12) which specifies an instruction (I10), and where instruction (I10) may have to enter a certain pipeline stage either before or after instruction (I9) enters or is allowed to enter a certain pipeline stage,
where the meaning of the bit-field (B12) may depend on the meaning of other bit-fields of the instruction format of any instruction of said instruction set,

where said microprocessor may have an instruction which, when it enters a certain pipeline stage, indicates that certain symbolic variables (Sk) are no longer used and that the values of symbolic variables (Sk) may be overwritten

where one or more instructions of the instruction set of said microprocessor may have an instruction format containing a bit-field (B14) which specifies whether or not a symbolic variable specified as instruction result may be allocated to the same register as the one to which one of the instruction operands is allocated

where one or more instructions of the instruction set of said microprocessor may have an instruction format containing a bit-field (B15) which specifies whether or not the value of a symbolic variable (sv4) specified as instruction result is the value of the definition address of said symbolic variable (sv4)

where one or more instructions of the instruction set of said microprocessor may have an instruction format containing a bit-field (B16) which specifies whether or not the value of a symbolic variable specified as instruction result is the value of a predication variable

where one or more instructions of the instruction set of said microprocessor may have an instruction format containing a bit-field (B17) which specifies whether or not the value of a symbolic variable specified as instruction result corresponds to a branch address

9. A method as claimed in claim 3.,
where one or more instructions of the instruction set of said microprocessor may have an instruction format containing a bit-field (B10) which indicates :
    - whether the value of an instruction operand is stored in a register file or not
    - or whether the value of an instruction operand has to be written into a register file or not
    - or whether the value of an instruction result has to be written into a register file or not

and where said instruction operand and said instruction result specify a symbolic variable, where the meaning of the bit-field (B10) may depend on the meaning of other bit-fields of the instruction format of any instruction of said instruction set,

where one or more instructions of the instruction set of said microprocessor may have an instruction format containing a bit-field (B11) which indicates whether a symbolic variable specified by an operand and/or result of an instruction (I7) is specified by an operand and/or result of another instruction (I8) or not, and where instruction (I8) may enter or may have to enter a certain pipeline stage either before or after instruction (I7) enters or is allowed to enter a certain pipeline stage,

where the meaning of the bit-field (B11) may depend on the meaning of other bit-fields of the instruction format of any instruction of said instruction set,

where one or more instructions of the instruction set of said microprocessor may have an instruction format containing a bit-field (B13) which indicates whether a symbolic variable specified by an instruction operand and/or an instruction result is data dependent or not,
where the meaning of the bit-field (B13) may depend on the meaning of other bit-fields of the instruction format of any instruction of said instruction set,

where at least one instruction (I9) of the instruction set of said microprocessor may have an instruction format containing a bit-field (B12) which specifies an instruction (I10), and where instruction (I10) may have to enter a certain pipeline stage either before or after instruction (I9) enters or is allowed to enter a certain pipeline stage,
where the meaning of the bit-field (B12) may depend on the meaning of other bit-fields of the instruction format of any instruction of said instruction set,

where said microprocessor may have an instruction which, when it enters a certain pipeline stage, indicates that certain symbolic variables (Sk) are no longer used and that the values of symbolic variables (Sk) may be overwritten

where one or more instructions of the instruction set of said microprocessor may have an instruction format containing a bit-field (B14) which specifies whether or not a symbolic variable specified as instruction result may be allocated to the same register as the one to which one of the instruction operands is allocated

where one or more instructions of the instruction set of said microprocessor may have an instruction format containing a bit-field (B15) which specifies whether or not the value of a symbolic variable (sv4) specified as instruction result is the value of the definition address of said symbolic variable (sv4)

where one or more instructions of the instruction set of said microprocessor may have an instruction format containing a bit-field (B16) which specifies whether or not the value of a symbolic variable specified as instruction result is the value of a predication variable

where one or more instructions of the instruction set of said microprocessor may have an instruction format containing a bit-field (B17) which specifies whether or not the value of a symbolic variable specified as instruction result corresponds to a branch address

10. A microprocessor as claimed in claim 4.,
where one or more instructions of the instruction set of said microprocessor may have an instruction format containing a bit-field (B10) which indicates :
- whether the value of an instruction operand is stored in a register file or not
- or whether the value of an instruction operand has to be written into a register file or not
- or whether the value of an instruction result has to be written into a register file or not

and where said instruction operand and said instruction result specify a symbolic variable,
where the meaning of the bit-field (B10) may depend on the meaning of other bit-fields of the instruction format of any instruction of said instruction set,

where one or more instructions of the instruction set of said microprocessor may have an instruction format containing a bit-field (B11) which indicates whether a symbolic variable specified by an operand and/or result of an instruction (I7) is specified by an operand and/or result of another instruction (I8) or not, and where instruction (I8) may enter or may have to enter a certain pipeline stage either before or after instruction (I7) enters or is allowed to enter a certain pipeline stage,
where the meaning of the bit-field (B11) may depend on the meaning of other bit-fields of the instruction format of any instruction of said instruction set,

where one or more instructions of the instruction set of said microprocessor may have an instruction format containing a bit-field (B13) which indicates whether a symbolic variable specified by an instruction operand and/or an instruction result is data dependent or not,
where the meaning of the bit-field (B13) may depend on the meaning of other bit-fields of the instruction format of any instruction of said instruction set,

where at least one instruction (I9) of the instruction set of said microprocessor may have an instruction format containing a bit-field (B12) which specifies an instruction (I10), and where instruction (I10) may have to enter a certain pipeline stage either before or after instruction (I9) enters or is allowed to enter a certain pipeline stage,
where the meaning of the bit-field (B12) may depend on the meaning of other bit-fields of the instruction format of any instruction of said instruction set,

where said microprocessor may have an instruction which, when it enters a certain pipeline stage, indicates that certain symbolic variables (Sk) are no longer used and that the values of symbolic variables (Sk) may be overwritten

where one or more instructions of the instruction set of said microprocessor may have an instruction format containing a bit-field (B14) which specifies whether or not a symbolic variable specified as instruction result may be allocated to the same register as the one to which one of the instruction operands is allocated where one or more instructions of the instruction set of said microprocessor may have an instruction format containing a bit-field (B15) which specifies whether or not the value of a symbolic variable (sv4) specified as instruction result is the value of the definition address of said symbolic variable (sv4)

where one or more instructions of the instruction set of said microprocessor may have an instruction format containing a bit-field (B16) which specifies whether or not the value of a symbolic variable specified as instruction result is the value of a predication variable

where one or more instructions of the instruction set of said microprocessor may have an instruction format containing a bit-field (B17) which specifies whether or not the value of a symbolic variable specified as instruction result corresponds to a branch address

11. A method as claimed in claim 5.,
where one or more instructions of the instruction set of said microprocessor may have an instruction format containing a bit-field (B10) which indicates :
- whether the value of an instruction operand is stored in a register file or not
- or whether the value of an instruction operand has to be written into a register file or not
- or whether the value of an instruction result has to be written into a register file or not

and where said instruction operand and said instruction result specify a symbolic variable,
where the meaning of the bit-field (B10) may depend on the meaning of other bit-fields of the instruction format of any instruction of said instruction set,

where one or more instructions of the instruction set of said microprocessor may have an instruction format containing a bit-field (B11) which indicates whether a symbolic variable specified by an operand and/or result of an instruction (I7) is specified by an operand and/or result of another instruction (I8) or not, and where instruction (I8) may enter or may have to enter a certain pipeline stage either before or after instruction (I7) enters or is allowed to enter a certain pipeline stage, where the meaning of the bit-field (B11) may depend on the meaning of other bit-fields of the instruction format of any instruction of said instruction set,

where one or more instructions of the instruction set of said microprocessor may have an instruction format containing a bit-field (B13) which indicates whether a symbolic variable specified by an instruction operand and/or an instruction result is data dependent or not,
where the meaning of the bit-field (B13) may depend on the meaning of other bit-fields of the instruction format of any instruction of said instruction set,

where at least one instruction (I9) of the instruction set of said microprocessor may have an instruction format containing a bit-field (B12) which specifies an instruction (I10), and where instruction (I10) may have to enter a certain pipeline stage either before or after instruction (I9) enters or is allowed to enter a certain pipeline stage,
where the meaning of the bit-field (B12) may depend on the meaning of other bit-fields of the instruction format of any instruction of said instruction set,

where said microprocessor may have an instruction which, when it enters a certain pipeline stage, indicates that certain symbolic variables (Sk) are no longer used and that the values of symbolic variables (Sk) may be overwritten

where one or more instructions of the instruction set of said microprocessor may have an instruction format containing a bit-field (B14) which specifies whether or not a symbolic variable specified as instruction result may be allocated to the same register as the one to which one of the instruction operands is allocated

where one or more instructions of the instruction set of said microprocessor may have an instruction format containing a bit-field (B15) which specifies whether or not the value of a symbolic variable (sv4) specified as instruction result is the value of the definition address of said symbolic variable (sv4)

where one or more instructions of the instruction set of said microprocessor may have an instruction format containing a bit-field (B16) which specifies whether or not the value of a symbolic variable specified as instruction result is the value of a predication variable

where one or more instructions of the instruction set of said microprocessor may have an instruction format containing a bit-field (B17) which specifies whether or not the value of a symbolic variable specified as instruction result corresponds to a branch address

12. A microprocessor as claimed in claim 6.,
where one or more instructions of the instruction set of said microprocessor may have an instruction format containing a bit-field (B10) which indicates :
- whether the value of an instruction operand is stored in a register file or not
- or whether the value of an instruction operand has to be written into a register file or not
- or whether the value of an instruction result has to be written into a register file or not

and where said instruction operand and said instruction result specify a symbolic variable,
where the meaning of the bit-field (B10) may depend on the meaning of other bit-fields of the instruction format of any instruction of said instruction set,

where one or more instructions of the instruction set of said microprocessor may have an instruction format containing a bit-field (B11) which indicates whether a symbolic variable specified by an operand and/or result of an instruction (I7) is specified by an operand and/or result of another instruction (I8) or not, and where instruction (I8) may enter or may have to enter a certain pipeline stage either before or after instruction (I7) enters or is allowed to enter a certain pipeline stage, where the meaning of the bit-field (B11) may depend on the meaning of other bit-fields of the instruction format of any instruction of said instruction set,

where one or more instructions of the instruction set of said microprocessor may have an instruction format containing a bit-field (B13) which indicates whether a symbolic variable specified by an instruction operand and/or an instruction result is data dependent or not,
where the meaning of the bit-field (B13) may depend on the meaning of other bit-fields of the instruction format of any instruction of said instruction set,

where at least one instruction (I9) of the instruction set of said microprocessor may have an instruction format containing a bit-field (B12) which specifies an instruction (I10), and where instruction (I10) may have to enter a certain pipeline stage either before or after instruction (I9) enters or is allowed to enter a certain pipeline stage,
where the meaning of the bit-field (B12) may depend on the meaning of other bit-fields of the instruction format of any instruction of said instruction set,

where said microprocessor may have an instruction which, when it enters a certain pipeline stage, indicates that certain symbolic variables (Sk) are no longer used and that the values of symbolic variables (Sk) may be overwritten

where one or more instructions of the instruction set of said microprocessor may have an instruction format containing a bit-field (B14) which specifies whether or not a symbolic variable specified as instruction result may be allocated to the same register as the one to which one of the instruction operands is allocated

where one or more instructions of the instruction set of said microprocessor may have an instruction format containing a bit-field (B15) which specifies whether or not the value of a symbolic variable (sv4) specified as instruction result is the value of the definition address of said symbolic variable (sv4)

where one or more instructions of the instruction set of said microprocessor may have an instruction format containing a bit-field (B16) which specifies whether or not the value of a symbolic variable specified as instruction result is the value of a predication variable where one or more instructions of the instruction set of said microprocessor may have an instruction format containing a bit-field (B17) which specifies whether or not the value of a symbolic variable specified as instruction result corresponds to a branch address